

Enhancing discovery in Spatial Data Infrastructures using search engines (#23406)

1

First submission

Editor guidance

Please submit by **15 Feb 2018** for the benefit of the authors (and your \$200 publishing discount).



Structure and Criteria

Please read the 'Structure and Criteria' page for general guidance.



Author notes

Have you read the author notes on the [guidance page](#)?



Raw data check

Review the raw data. Download from the location [described by the author](#).



Image check

Check that figures and images have not been inappropriately manipulated.

Privacy reminder: If uploading an annotated PDF, remove identifiable information to remain anonymous.

Files

Download and review all files from the [materials page](#).

6 Figure file(s)


2 Latex file(s)



Structure your review

The review form is divided into 5 sections.
Please consider these when composing your review:

1. BASIC REPORTING
2. EXPERIMENTAL DESIGN
3. VALIDITY OF THE FINDINGS
4. General comments
5. Confidential notes to the editor






 You can also annotate this PDF and upload it as part of your review

When ready [submit online](#).





Editorial Criteria

Use these criteria points to structure your review. The full detailed editorial criteria is on your [guidance page](#).





BASIC REPORTING

-  Clear, unambiguous, professional English language used throughout.
-  Intro & background to show context. Literature well referenced & relevant.
-  Structure conforms to [PeerJ standards](#), discipline norm, or improved for clarity.
-  Figures are relevant, high quality, well labelled & described.
-  Raw data supplied (see [PeerJ policy](#)).

EXPERIMENTAL DESIGN

-  Original primary research within [Scope of the journal](#).
-  Research question well defined, relevant & meaningful. It is stated how the research fills an identified knowledge gap.
-  Rigorous investigation performed to a high technical & ethical standard.
-  Methods described with sufficient detail & information to replicate.

VALIDITY OF THE FINDINGS

-  Impact and novelty not assessed. Negative/inconclusive results accepted. *Meaningful* replication encouraged where rationale & benefit to literature is clearly stated.
-  Data is robust, statistically sound, & controlled.
-  Conclusions are well stated, linked to original research question & limited to supporting results.
-  Speculation is welcome, but should be identified as such.



The best reviewers use these techniques

Tip

Support criticisms with evidence from the text or from other sources

Example

Smith et al (J of Methodology, 2005, V3, pp 123) have shown that the analysis you use in Lines 241-250 is not the most appropriate for this situation. Please explain why you used this method.

Give specific suggestions on how to improve the manuscript

Your introduction needs more detail. I suggest that you improve the description at lines 57- 86 to provide more justification for your study (specifically, you should expand upon the knowledge gap being filled).

Comment on language and grammar issues

The English language should be improved to ensure that an international audience can clearly understand your text. Some examples where the language could be improved include lines 23, 77, 121, 128 - the current phrasing makes comprehension difficult.

Organize by importance of the issues, and number your points

- 1. Your most important issue*
- 2. The next most important item*
- 3. ...*
- 4. The least important points*

Please provide constructive criticism, and avoid personal opinions

I thank you for providing the raw data, however your supplemental files need more descriptive metadata identifiers to be useful to future readers. Although your results are compelling, the data analysis should be improved in the following ways: AA, BB, CC

Comment on strengths (as well as weaknesses) of the manuscript

I commend the authors for their extensive data set, compiled over many years of detailed fieldwork. In addition, the manuscript is clearly written in professional, unambiguous language. If there is a weakness, it is in the statistical analysis (as I have noted above) which should be improved upon before Acceptance.

Enhancing discovery in Spatial Data Infrastructures using search engines

Paolo Corti ^{Corresp., 1}, Athanasios Tom Kralidis ², Benjamin Lewis ¹

¹ Center for Geographic Analysis, Harvard University, Cambridge, Massachusetts, United States

² Open Source Geospatial Foundation, Beaverton, Oregon, United States

Corresponding Author: Paolo Corti
Email address: pcorti@gmail.com

A Spatial Data Infrastructure (SDI) is a framework of geospatial data, metadata, users and tools intended to provide an efficient and flexible way to use spatial information. One of the key software components of an SDI is the catalogue service which is needed to discover, query, and manage the metadata. Catalogue services in an SDI are typically based on the Open Geospatial Consortium (OGC) Catalogue Service for the Web (CSW) standard which defines common interfaces for accessing the metadata information.

A search engine is a software system capable of supporting fast and reliable search, which may use “any means necessary” to get users to the resources they need quickly and efficiently. These techniques may include full text search, natural language processing, weighted results, fuzzy tolerance results, faceting, hit highlighting, recommendations and many others. In this paper we present an example of a search engine being added to an SDI to improve search against large collections of geospatial datasets.

In work funded by the National Endowment for the Humanities, the Centre for Geographic Analysis (CGA) at Harvard University re-engineered the search component of its public domain SDI (WorldMap) which is based on the GeoNode platform. In the process the CGA developed Harvard Hypermap, a map services registry and search platform independent from WorldMap, which is built on a search engine.

The goal of Hypermap is to provide a framework for building and maintaining a comprehensive registry of web map services, and in addition, encourage the development of clients with modern search capabilities such as spatial and temporal faceting and instant previews. Behind the scenes Hypermap scalably harvests OGC and Esri service metadata from distributed servers, organizes that information, and pushes it to a search engine. The system monitors services for reliability and uses that to improve search. End users are also able to search the SDI metadata using standard interfaces provided by the internal CSW catalogue, and at the same time benefit from the enhanced search possibilities provided by an advanced search engine. Hypermap is built on an open source software stack.

1 Enhancing discovery in Spatial Data 2 Infrastructures using search engines

3 Paolo Corti¹, Athanasios Tom Kralidis², and Benjamin Lewis¹

4 ¹Center for Geographic Analysis, Harvard University, Cambridge MA USA

5 ²Open Source Geospatial Foundation, Beaverton OR USA

6 Corresponding author:

7 Paolo Corti¹

8 Email address: pcorti@gmail.com

9 ABSTRACT

10 A Spatial Data Infrastructure (SDI) is a framework of geospatial data, metadata, users and tools intended
11 to provide an efficient and flexible way to use spatial information. One of the key software components of
12 an SDI is the catalogue service which is needed to discover, query, and manage the metadata. Catalogue
13 services in an SDI are typically based on the Open Geospatial Consortium (OGC) Catalogue Service for
14 the Web (CSW) standard which defines common interfaces for accessing the metadata information.

15 A search engine is a software system capable of supporting fast and reliable search, which may use “any
16 means necessary” to get users to the resources they need quickly and efficiently. These techniques may
17 include full text search, natural language processing, weighted results, fuzzy tolerance results, faceting,
18 hit highlighting, recommendations and many others. In this paper we present an example of a search
19 engine being added to an SDI to improve search against large collections of geospatial datasets.

20 In work funded by the National Endowment for the Humanities, the Centre for Geographic Analysis (CGA)
21 at Harvard University re-engineered the search component of its public domain SDI (WorldMap) which is
22 based on the GeoNode platform. In the process the CGA developed Harvard Hypermap, a map services
23 registry and search platform independent from WorldMap, which is built on a search engine.

24 The goal of Hypermap is to provide a framework for building and maintaining a comprehensive registry of
25 web map services, and in addition, encourage the development of clients with modern search capabilities
26 such as spatial and temporal faceting and instant previews. Behind the scenes Hypermap scalably
27 harvests OGC and Esri service metadata from distributed servers, organizes that information, and pushes
28 it to a search engine. The system monitors services for reliability and uses that to improve search. End
29 users are also able to search the SDI metadata using standard interfaces provided by the internal CSW
30 catalogue, and at the same time benefit from the enhanced search possibilities provided by an advanced
31 search engine. Hypermap is built on an open source software stack.

32 INTRODUCTION

33 A Spatial Database Infrastructure (SDI) typically stores a large collection of metadata. While the Open
34 Geospatial Consortium (OGC) recommends the use of the Catalogue Service for the Web (CSW) standard
35 to query these metadata, several important benefits can be obtained by pairing the CSW with a search
36 engine platform within the SDI software stack.

37 SDI, Interoperability, and Standards

38 A Spatial Data Infrastructure (SDI) is a framework of geospatial data, metadata, users and tools which
39 provides a mechanism for publishing and updating geospatial information. An SDI provides the archi-
40 tectural underpinnings for the discovery, evaluation, and use of geospatial information (Infrastructures,
41 2004; Goodchild et al., 2007; Masó et al., 2012). SDIs are typically distributed in nature, and connected
42 by disparate computing platforms and client/server design patterns.

43 A critical principle of an SDI is interoperability which can be defined as the ability of a system or
44 components in a system to provide information sharing and inter-application cooperative process control
45 through a mutual understanding of request and response mechanisms embodied in standards.

46 Standards (formal, de facto, community) provide three primary benefits for geospatial information:
47 a) portability: use and reuse of information and applications, b) interoperability: multiple system
48 information exchange and c) maintainability: long term updating and effective use of a resource (Groot
49 and McLaughlin, 2000). The OGC standards baseline has traditionally provided core standards definitions
50 to major SDI activities. Along with other standards bodies (IETF, ISO, OASIS) and de facto / community
51 efforts (Open Source Geospatial Foundation [OSGeo], etc.), OGC standards provide broadly accepted,
52 mature specifications, profiles, and best practices (Kralidis, 2009).

53 **Metadata search in an SDI and CSW**

54 An SDI can contain a large number of geospatial datasets which may grow in number over time. The
55 difficulty of finding a needle in such a haystack means a more effective metadata search mechanism is
56 called for. Metadata is data about data, describe the content, quality, condition, and other characteristics of
57 data in order to ease the search and understanding of data (Nogueras-Iso et al., 2005). Metadata standards
58 define a way to provide homogeneous information about the identification, the extent, the spatial and
59 temporal aspects, the content, the spatial reference, the portrayal, distribution and other properties of
60 digital geographic data and services (ISO, 2014).

61 Ease of data discovery is a critical measure of the effectiveness of an SDI. The OGC Catalogue
62 interface standards (Catalogue Service for the Web - CSW) specify the interfaces and bindings, as well
63 as a framework for defining the application profiles required to publish and access digital catalogues of
64 metadata for geospatial data and services. (Consortium, 2016; Nebert et al., 2005; Rajabifard et al., 2009).

65 Based on the Dublin Core metadata information model, CSW supports broad interoperability around
66 discovering geospatial data and services spatially, non-spatially, temporally, and via keywords or free
67 text. CSW supports application profiles which allow for information communities to constrain and/or
68 extend the CSW specification to satisfy specific discovery requirements and to realize tighter coupling
69 and integration of geospatial data and services. The CSW ISO Application Profile is an example of a
70 standard for geospatial data search which follows ISO geospatial metadata standards.

71 **Need for a search engine within an SDI**

72 Search workflow and user experience are a vital part of modern web-based applications. Numerous types
73 of web application such as Content Management Systems (CMS), wikis, data delivery frameworks, all
74 can benefit from improved data discovery. Same applies to SDI. Furthermore, in the Big Data era, more
75 powerful mechanisms are needed to return relevant content to the users from very large collections of data
76 (Tsinaraki and Schade, 2016).

77 In the last few years, content-driven platforms have delegated the task of search optimization to
78 specific frameworks known as search engines. Rather than implementing a custom search logic, these
79 platforms now often add a search engine in the stack to improve search. Apache Solr (Solr, 2011) and
80 Elasticsearch (Elasticsearch, 2015), two popular open source search engine web platforms, both based on
81 Apache Lucene (Cutting et al., 2004), are commonly used in typical web application stacks to support
82 complex search criteria, faceting, results highlighting, query spell-check, relevance tuning and more
83 (Smiley et al., 2015). As for CMS's, SDI search can dramatically benefit from such platforms as well.

84 ***How a search engine works***

85 Typically the way a search engine works can be split into two distinct phases: indexing and searching.
86 During the indexing phase, all of the documents (metadata, in the SDI context) that must be searched are
87 scanned, and a list of search terms (an index) is built. For each search term, the index keeps track of the
88 identifiers of the documents that contain the search term. During the searching phase only the index is
89 looked at, and a list of the documents containing the given search term is quickly returned to the client.
90 This indexed approach makes a search engine extremely fast in outputting results. On top of this, a search
91 engine provides many other useful search related features, improving dramatically the experience of users.

92 ***Improvements in a SDI with a search engine***

93 There are numerous opportunities to enhance the functionality of the CSW specification and subsequent
94 server implementations by specifying standard search engine functionality as enhancements to the standard.
95 A search engine is extremely fast and scalable: by building and maintaining its indexed structure of
96 the content, it can return results much faster and scale much better than a traditional CSW based on a
97 relational database. While a CSW can search metadata with a full text approach, with a search engine it is

98 possible to extend the full text search with features such as language stemming, thesaurus and synonyms,
99 hit highlighting, wildcard matches and other “fuzzy” matching techniques. Another key advantage is
100 that search engines can provide relevancy weights to likely matches, allowing for much finer tuning of
101 search results. CSW does not easily emit facets or facet counts as part of search results. Search engine
102 facets however, can be based on numerous classification schemes, such as named geography, date and
103 time extent, keywords, etc. and can be used to enable interactive feedback mechanisms which help users
104 define and refine their searches effectively.

105 BACKGROUND

106 WorldMap (Team, 2010) is an open source SDI and Geospatial Content Management System (GeoCMS)
107 platform developed by the CGA to lower the barrier for scholars who wish to explore, visualize, edit and
108 publish geospatial information (Guan et al., 2012). Registered users are able to upload geospatial content,
109 in the form of vector or raster datasets (layers), and combine them with existing layers to create maps.
110 Existing layers can be layers uploaded by other users and layers provided by external map services.

111 WorldMap is a web application built on top of the GeoNode open source mapping platform (Team,
112 2009c), and since 2012 has been used by more than 20,000 registered users to upload about 30,000 layers
113 and to create some 7,000 web maps.

114 GeoNode is based on the following components, all open source and designed around OGC standards
115 (Figure 1):

- 116 • a JavaScript client, GeoExplorer (Team, 2009b), based on OpenLayers (Team, 2005b) and ExtJS
117 (Team, 2007a)
- 118 • a map server engine based on GeoServer (Team, 2001a)
- 119 • a tile cache server based on GeoWebCache (Team, 2002)
- 120 • a spatial database implemented with PostgreSQL (Team, 1997) and PostGIS (Team, 2001b)
- 121 • a catalogue based on pycsw (pycsw Development Core Team, 2011) or GeoNetwork (Team, 2007b)
- 122 • a web application, developed with Django (Team, 2005a), a Python web framework, which orches-
123 trates all of the previous components

124 WorldMap allows users to build maps using its internal catalogue of layers (local layers) combined
125 with layers from external map services (remote layers), for a total of about 200,000 layers. WorldMap
126 users can have trouble finding useful and reliable layers with such a large number. A system was needed
127 to enable fast, scalable search which generally returns to the user the most reliable and useful layers
128 known to the system.

129 RESULTS AND DISCUSSION

130 Hypermap

131 In 2014 CGA started the design and development of Hypermap Registry (Hypermap) (CGA, 2017) to
132 provide to WorldMap users a more effective search experience. Hypermap is an application that manages
133 OGC web services (such as WMS, WMTS, CSW Capabilities service metadata) and Esri REST endpoints
134 and in addition supports map service discovery (Chen et al., 2011), crawling (Bone et al., 2016; Li et al.,
135 2010), harvesting, and uptime statistics gathering for services and layers. All of the services and endpoints
136 are parsed in order to detect the exposed layers, including the ones exposed by the WorldMap instance. By
137 repeating this operation multiple times, using asynchronous processing based on a task queue, metadata
138 and uptime statistics are harvested for service and layers. This way it is possible to return to end users
139 scored results, with layers being more reliable at the top. When possible the layers are cached by a map
140 cache engine, in order to provide faster and more reliable tiles to the end users. A CSW catalogue is
141 included; this way metadata can be searched by clients using the OGC standard for broad interoperability.

142 While with CSW endpoint external clients such as QGIS and ArcGIS Desktop can search within the
143 layers catalogue using the OGC CSW standard, an improved REST API which exposes extended search
144 features is needed. For this purpose a search engine was included in the Hypermap stack.

145 The Hypermap stack is composed of the following components:

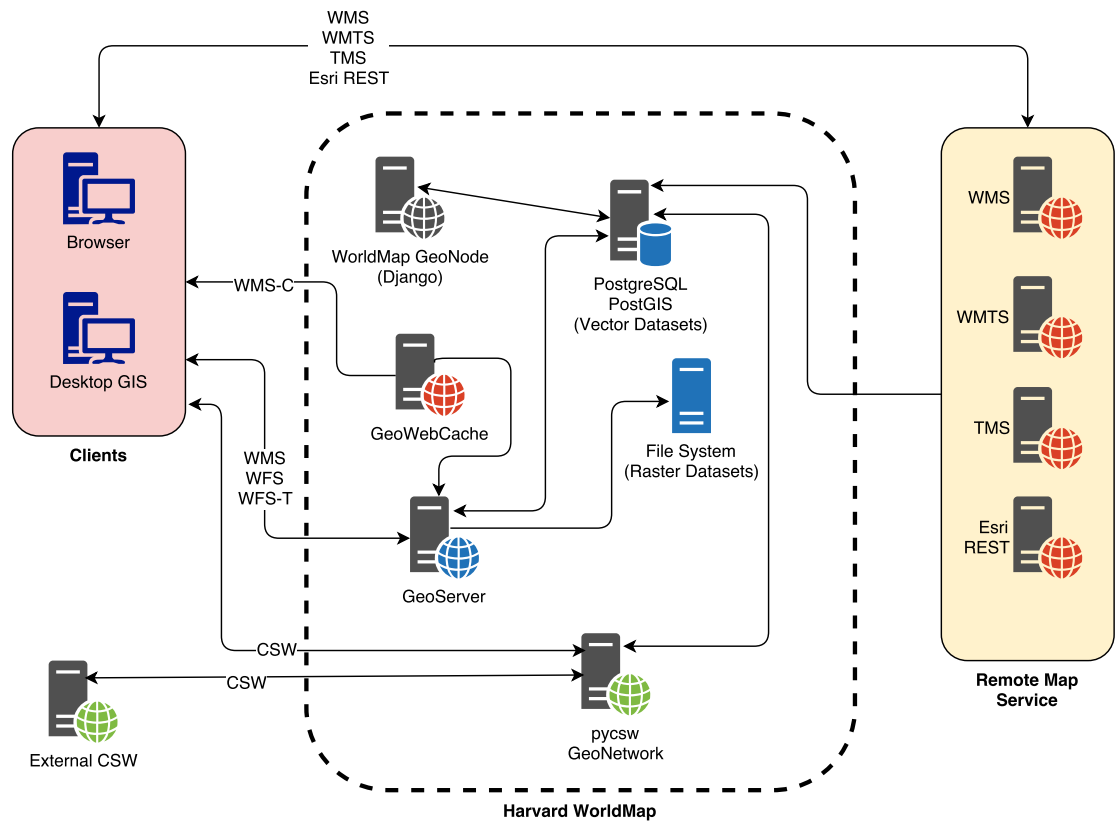


Figure 1. The WorldMap SDI architecture

- 146 • A web application developed with Django
- 147 • A relational database based on PostgreSQL
- 148 • A map cache implemented with MapProxy (Tonnhofner and Helle, 2014)
- 149 • An OGC catalogue based on pycsw
- 150 • A task queue based on Celery (Team, 2009a) and RabbitMQ (Team, 2007c)
- 151 • A search engine based on Solr or Elasticsearch, both based on Lucene
- 152 • An API designed around Swagger (Team, 2011)

153 Hypermap search engine is synchronized with the needed content from its relational database using
 154 tasks in a task queue.

155 WorldMap interacts with Hypermap search engine by querying the Swagger API, which wraps the
 156 search engine API. The WorldMap users can search the existing layers metadata (both local WorldMap
 157 layers and remote service layers) filtering on keywords, source, layer type, map extent and date range
 158 (Corti and Lewis, 2017). The Hypermap API returns results from the search engine with a JSON output,
 159 and a tabular view and a spatial view (based on spatial facets) are returned to the user browser (Figure 2).

160 **WorldMap improvements with the search engine**

161 By pairing the CSW catalogue with a search engine, the metadata search in the WorldMap SDI yields
 162 several major benefits.

163 **Fast results**

164 By having the metadata content indexed in a search engine, metadata are returned extremely fast to the
 165 client. On average, a full text query using the search engine (Solr) is about 20/30 times faster than the
 166 corresponding query to the CSW catalogue (pycsw) with an optimized relational database.



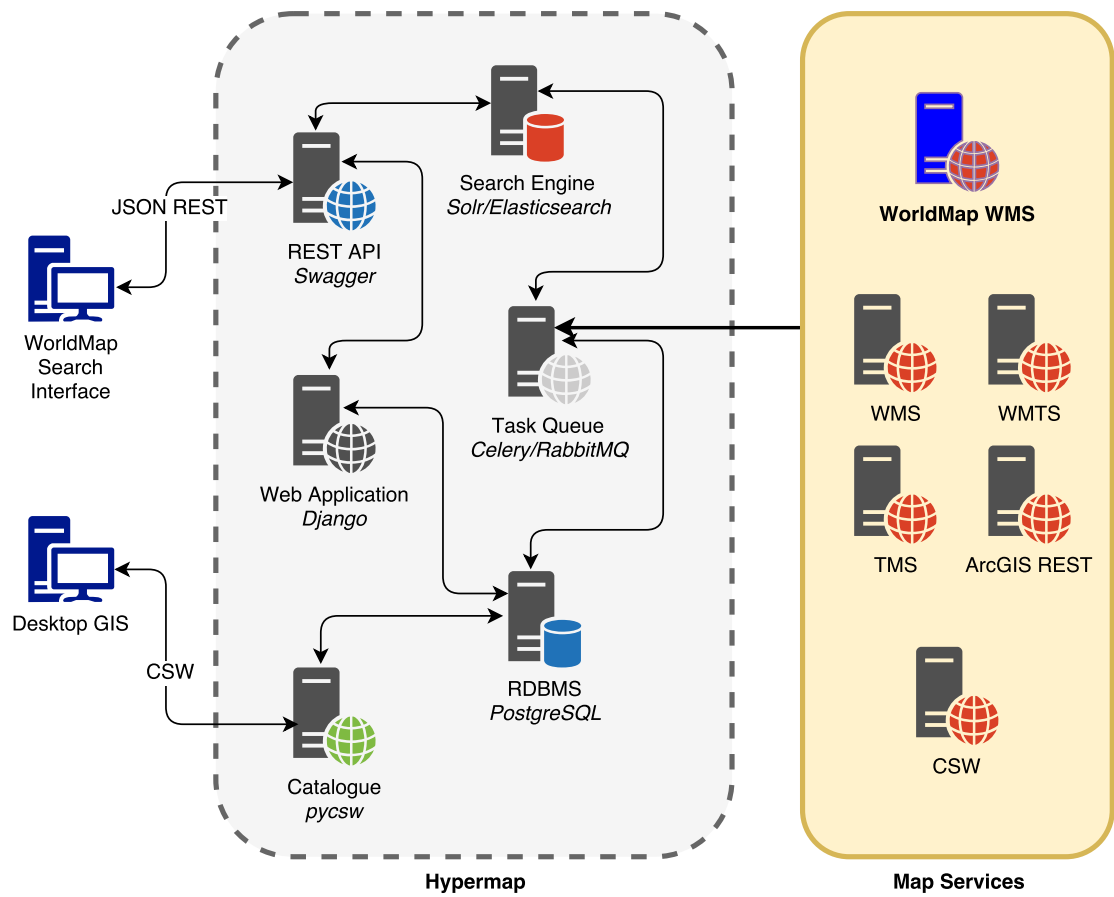


Figure 2. WorldMap/Hypermap interaction

167 **Scalability**

168 From a software engineering perspective, search engines are highly scalable and replicable, thanks to their
 169 shardable architecture. Such systems are capable of providing interactive query access to collections of
 170 spatio-temporal objects containing billions of features (Kakkar and Lewis, 2017; Kakkar et al., 2017).

171 **Clean API**

172 Query to the search engine API tends to be much simpler than XML queries to the CSW catalogue,
 173 specially when crafting advanced search requests (spatial, non-spatial, temporal, etc. . .). Same for output:
 174 JSON output from search engine API provides a more compact representation of search results enabling
 175 better performance and making the output more readable (Figure 3, Figure 4).

176 **Synonyms, text stemming**

177 Crucially, search engines are good at handling the ambiguities of natural languages, thanks to stop words
 178 (words filtered out during the processing of text), stemming (ability to detect words derived from a
 179 common root), synonyms detection, and controlled vocabularies such as thesauri and taxonomies. It
 180 is possible to do phrase searches and proximity searches (search for a phrase containing two different
 181 words separated by a specified number of words). Because of features like these, keyword queries using
 182 the Hypermap search engine endpoint typically returns more results than an equivalent query using the
 183 Hypermap CSW. For example doing a full text search for the keyword “libraries” returns more results
 184 from the search engine because it includes variations and synonyms of the original term like “library”,
 185 “bibliotheca”, “repository”, “repositories” in the returned results.

186 **Relevancy**

187 Results can be ranked, providing a way to return results to users with the more relevant ones closer to
 188 the top. This is very useful to detect the most significant metadata for a given query. Weights can be

```

Request:--
<?xml version="1.0" ?>
<csw:GetRecords maxRecords="10" outputFormat="application/xml" outputSchema="http://www.opengis.net/cat/csw/2.0.2" resultType="results"
service="CSW" version="2.0.2" xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2
http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd">
  <csw:Query typeNames="csw:Record">
    <csw:ElementSetName>full</csw:ElementSetName>
    <csw:Constraint version="1.1.0">
      <ogc:Filter>
        <ogc:PropertyIsLike escapeChar="\" singleChar="\" wildCard="%">
          <ogc:PropertyName>csw:AnyText</ogc:PropertyName>
          <ogc:Literal>libraries in boston</ogc:Literal>
        </ogc:PropertyIsLike>
      </ogc:Filter>
    </csw:Constraint>
  </csw:Query>
</csw:GetRecords>

Response:--
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- pyocsw 2.1-dev-20161019 -->
<csw:GetRecordsResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dct="http://purl.org/dc/terms/" xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.0.2" xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2 http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd">
  <csw:SearchStatus timestamp="2018-01-10T23:26:57Z"/>
  <csw:SearchResults nextRecord="0" numberOfRecordsMatched="1" numberOfRecordsReturned="1" recordSchema="http://www.opengis.net/cat/csw/2.0.2"
elementSetName="full">
    <csw:Record>
      <dc:identifier>b8482b92-c0b3-11e4-824f-22000aeecbbb</dc:identifier>
      <dc:title>Libraries in Boston</dc:title>
      <dc:alternative>geonode:alex_zku</dc:alternative>
      <dc:modified>2018-01-09T14:01:27Z</dc:modified>
      <dc:abstract>Showing the different libraries in the Boston area</dc:abstract>
      <dc:type>dataset</dc:type>
      <dc:format>Hypermap:WorldMap</dc:format>
      <dc:source>http://worldmap.harvard.edu/geoserver/geonode/geonode:alex_zku/wms?</dc:source>
      <dc:relation>2a96b71c-96b2-4432-b31f-219c45f3fc52</dc:relation>
      <dc:references scheme="Hypermap:WorldMap">http://worldmap.harvard.edu/geoserver/geonode/geonode:alex_zku/wms?</dc:references>
      <dc:references scheme="OGC:WMTS">http://localhost:8000/http://worldmap.harvard.edu/geoserver/geonode/geonode:alex_zku/wms?</dc:references>
      <ows:BoundingBox crs="http://www.opengis.net/def/crs/EPSSG/0/4326" dimensions="2">
        <ows:LowerCorner>-1.0 -1.0</ows:LowerCorner>
        <ows:UpperCorner>0.0 0.0</ows:UpperCorner>
      </ows:BoundingBox>
    </csw:Record>
  </csw:SearchResults>
</csw:GetRecordsResponse>

```

Figure 3. CSW Request and Response

189 assigned by specifying boosts (weighted factors) for each field.

190 Facets

191 Another important search engine feature useful for searching the WorldMap metadata catalogue is faceted
 192 search. Faceting is the arrangement of search results in categories based on indexed terms. This capability
 193 makes it possible for example, to provide an immediate indication of the number of times that common
 194 keywords are contained in different metadata documents. A typical use case is with metadata categories,
 195 keywords and regions. Thanks to facets, the user interface of an SDI catalogue can display counts for
 196 documents by category, keyword or region (Figure 5).

197 Search engines can also support temporal and spatial faceting, two features that are extremely useful
 198 for browsing large collections of geospatial metadata. Temporal faceting can display the number of
 199 metadata documents by date range as a kind of histogram. Spatial faceting can provide a spatial surface
 200 representing the distribution of layers or features across an area of interest. In Figure 6 a heatmap is
 201 generated by spatial faceting which shows the distribution of layers in the WorldMap SDI for a given
 202 geographic region (Figure 6).

203 Other features

204 In addition, it is possible to use regular expressions, wildcard search, and fuzzy search to provide results
 205 for a given term and its common variations. It is also possible to support boolean queries: a user is able to
 206 search results using terms and boolean operators such as AND, OR, NOT and hit highlighting can provide
 207 immediate search term suggestions to the user searching a text string in metadata.

208 CONCLUSIONS

209 While the CSW 3.0.0 standard provides improvements to address mass market search/discovery, the
210 benefits of search engine implementations combined with broad interoperability of the CSW standard
211 presents a great opportunity to enhance the CSW standard. The authors hope that such an approach
212 eventually becomes formalized as a CSW Application Profile or Best Practice in order to achieve maximum
213 benefit and adoption in SDI activities. This will allow CSW implementations to make better use of search
214 engine methodologies for improving the user search experience in SDI workflows.

215 The authors would also like to share the work with the OGC CSW community in support of the
216 evolution of the CSW specification. Given recent developments on the OGC WFS 3.0 standard (RESTful
217 design patterns, JSON, etc.), there is also an opportunity for CSW to evolve in alignment with WFS 3.0 in
218 support of the principles of the W3C Spatial Data on the Web Best Practices (Group, 2017) in a similar
219 manner as presented by the work in this paper.

220 Harvard Hypermap aims to provide a FOSS solution using modern approaches to realize a highly
221 scalable, flexible and robust geospatial registry and catalogue/search platform while achieving broad
222 interoperability via open standards. In addition, pycsw is planning for dedicated Elasticsearch/Solr support
223 as part of a future release to support the use of search engines as backend stores to the CSW standard.

224 ACKNOWLEDGEMENTS

225 The authors thanks all the contributors to the Hypermap and WorldMap platform source code, particularly:
226 Matt Bertrand, Simone Dalmaso, Alessio Fabiani, Jorge Martínez Gómez, Wendy Guan, Jeffrey Johnson,
227 Devika Kakkar, Jude Mwenda, Ariel Núñez, Luis Pallares, David Smiley, Charles Thao, Angelos Tzotsos,
228 Mingda Zhang

229 REFERENCES

- 230 Bone, C., Ager, A., Bunzel, K., and Tierney, L. (2016). A geospatial search engine for discovering
231 multi-format geospatial data across the web. *International Journal of Digital Earth*, 9(1):47–62.
- 232 CGA, H. (2017). Hypermap registry github repository.
- 233 Chen, N., Chen, Z., Hu, C., and Di, L. (2011). A capability matching and ontology reasoning method for
234 high precision ogc web service discovery. *International Journal of Digital Earth*, 4(6):449–470.
- 235 Consortium, O. G. (2016). Catalogue Service. <http://www.opengeospatial.org/standards/cat/>. [Online; accessed 13-April-2017].
- 236
- 237 Corti, P. and Lewis, B. (2017). Making temporal search more central in spatial data infrastructures. *ISPRS*
238 *Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 93–95.
- 239 Cutting, D. et al. (2004). Lucene. <https://lucene.apache.org/>.
- 240 Elasticsearch, B. (2015). Elasticsearch. <https://www.elastic.co/>.
- 241 Goodchild, M. F., Fu, P., and Rich, P. (2007). Sharing geographic information: an assessment of the
242 geospatial one-stop. *Annals of the Association of American Geographers*, 97(2):250–266.
- 243 Groot, R. and McLaughlin, J. D. (2000). *Geospatial data infrastructure: concepts, cases, and good*
244 *practice*. Oxford university press Oxford.
- 245 Group, O. W. W. (2017). Spatial data on the web best practices. <https://www.w3.org/TR/sdw-bp/>.
- 246
- 247 Guan, W. W., Bol, P. K., Lewis, B. G., Bertrand, M., Berman, M. L., and Blossom, J. C. (2012).
248 Worldmap—a geospatial framework for collaborative research. *Annals of GIS*, 18(2):121–134.
- 249 Infrastructures, D. S. D. (2004). the sdi cookbook. *GSDI/Nebert*.
- 250 ISO, I. (2014). 19115-1: 2014 geographic information-metadata-part 1: Fundamentals. *International*
251 *Standards Organisation, Geneva, Switzerland*.
- 252 Kakkar, D. and Lewis, B. (2017). Building a billion spatio-temporal object search and visualization
253 platform. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages
254 97–100.
- 255 Kakkar, D., Lewis, B., Smiley, D., and Nunez, A. (2017). The billion object platform (bop): a system
256 to lower barriers to support big, streaming, spatio-temporal data sources. In *Free and Open Source*
257 *Software for Geospatial (FOSS4G) Conference Proceedings*, volume 17, page 15.
- 258 Kralidis, A. T. (2009). Geospatial web services: The evolution of geospatial data infrastructure. In *The*
259 *Geospatial Web*, pages 223–228. Springer.

- 260 Li, W., Yang, C., and Yang, C. (2010). An active crawler for discovering geospatial web services and
261 their distribution pattern—a case study of ogc web map service. *International Journal of Geographical*
262 *Information Science*, 24(8):1127–1147.
- 263 Masó, J., Pons, X., and Zabala, A. (2012). Tuning the second-generation sdi: theoretical aspects and real
264 use cases. *International Journal of Geographical Information Science*, 26(6):983–1014.
- 265 Nebert, D., Whiteside, A., and Vretanos, P. (2005). Ogc catalogue services specification. *Open Geospatial*
266 *Consortium Inc.*
- 267 Nogueras-Iso, J., Zarazaga-Soria, F. J., and Muro-Medrano, P. R. (2005). Geographic information
268 metadata for spatial data infrastructures. *Resources, Interoperability and Information Retrieval*.
- 269 pycsw Development Core Team (2011). pycsw. <http://pycsw.org/>.
- 270 Rajabifard, A., Kalantari, M., and Binns, A. (2009). Sdi and metadata entry and updating tools. *SDI*
271 *convergence*, 121.
- 272 Smiley, D., Pugh, E., Parisa, K., and Mitchell, M. (2015). *Apache Solr enterprise search server*. Packt
273 Publishing Ltd.
- 274 Solr, A. (2011). Apache solr. <http://lucene.apache.org/solr/>.
- 275 Team, C. D. C. (2009a). Celery. <http://www.celeryproject.org/>.
- 276 Team, D. D. C. (2005a). Django. <https://www.djangoproject.com/>.
- 277 Team, E. D. C. (2007a). Extjs. <https://www.sencha.com/products/extjs/>.
- 278 Team, G. D. C. (2001a). Geoserver. <http://geoserver.org/>.
- 279 Team, G. D. C. (2002). Geowebcache. <http://geowebcache.org/>.
- 280 Team, G. D. C. (2007b). Geonetwork. <https://geonetwork-opensource.org/>.
- 281 Team, G. D. C. (2009b). Geoexplorer. <http://suite.boundlessgeo.com/docs/latest/>.
- 282 Team, G. D. C. (2009c). Geonode. <http://geonode.org/>.
- 283 Team, O. D. C. (2005b). Openlayers. <https://openlayers.org/>.
- 284 Team, P. D. C. (1997). Postgresql. <https://www.postgresql.org/>.
- 285 Team, P. D. C. (2001b). Postgis. <https://postgis.net/>.
- 286 Team, R. D. C. (2007c). Rabbitmq. <https://www.rabbitmq.com/>.
- 287 Team, S. D. C. (2011). Swagger. <https://swagger.io/>.
- 288 Team, W. D. C. (2010). Worldmap. <http://worldmap.harvard.edu/>.
- 289 Tonnhofer, O. and Helle, D. (2014). Mapproxy—open source proxy for geospatial data. [https://](https://mapproxy.org/)
290 mapproxy.org/.
- 291 Tsinaraki, C. and Schade, S. (2016). Big data—a step change for sdi? *International Journal*, 11:09–19.



```

Request:~
~
http://worldmap.harvard.edu/solr/hypermaph/select?indent=on&q=_text_%22libraries%20in%20boston%22&wt=json~
~
Response:~
~
{~
  .."responseHeader":{~
    ....."status":0,~
    ....."QTime":1,~
    ....."params":{~
      ....."q": "_text_: \"libraries in boston\"",~
      ....."indent": "on",~
      ....."wt": "json"},~
    ....."response": {"numFound": 1, "start": 0, "docs": [~
      ....."{~
        ....."abstract": "Showing the different libraries in the Boston area",~
        ....."layer_username": "ambilaa",~
        ....."reliability": 97.96954314720813,~
        ....."layer_originator": "worldmap.harvard.edu",~
        ....."recent_reliability": 100.0,~
        ....."availability": "OnLine",~
        ....."min_x": -1.0,~
        ....."uuid": "b8482b92-c0b3-11e4-824f-22000aeecbbb",~
        ....."title": "Libraries in Boston",~
        ....."domain_name": "worldmap.harvard.edu",~
        ....."id": 190905,~
        ....."is_valid": true,~
        ....."location": {"layerInfoPage": "\ /registry/hypermaph/layer/b8482b92-c0b3-11e4-824f-22000aeecbbb/\\"},~
        ....."service_type": "Hypermaph:WorldMap",~
        ....."layer_category": ["Place Locations"],~
        ....."max_x": 0.0,~
        ....."max_y": 0.0,~
        ....."layer_datetype": "From Metadata",~
        ....."layer_id": 190905,~
        ....."bbox": "ENVELOPE(-1.000000,0.000000,0.000000,-1.000000)",~
        ....."last_status": true,~
        ....."is_public": true,~
        ....."min_y": -1.0,~
        ....."layer_date": "2016-11-15T21:55:03Z",~
        ....."name": "geonode:alex_zku",~
        ....."url": "http://worldmap.harvard.edu/geoserver/geonode/geonode:alex_zku/wms?",~
        ....."type": "Layer",~
        ....."area": 1.0,~
        ....."tile_url": "/registry/hypermaph/layer/190905/map/wmts/geonode:alex_zku/default_grid/{z}/{x}/{y}.png",~
        ....."srs": ["EPSG:4326",~
          ....."EPSG:900913",~
          ....."EPSG:3857"],~
        ....."centroid_y": [-0.5],~
        ....."centroid_x": [-0.5],~
        ....."service_id": 23,~
        ....."_version_": 1589125870789853184,~
        ....."timestamp": "2018-01-09T14:35:28.508Z"}~
      ....."}]~
    ....."}}~
  ....."}}~
}~

```

Figure 4. Search Engine Request and Response

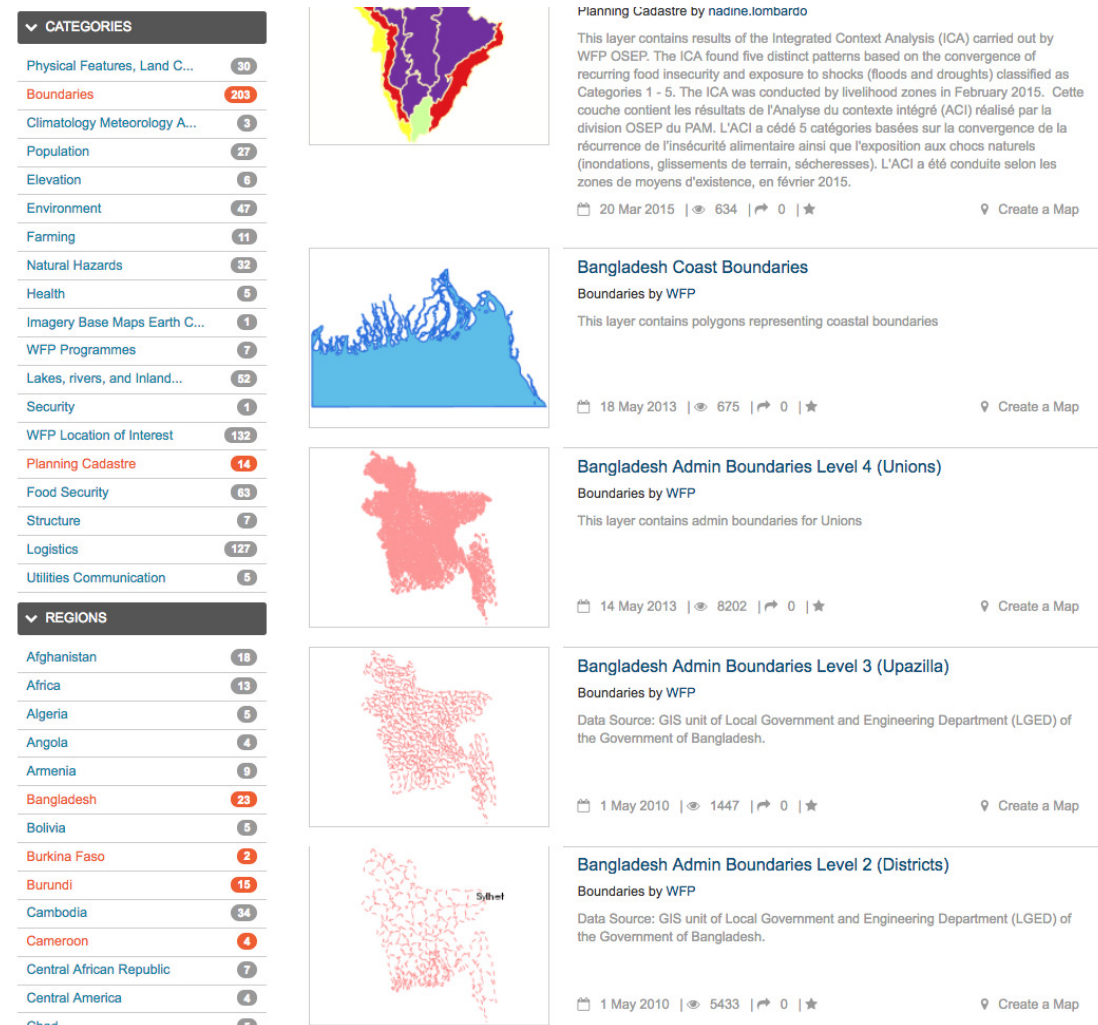


Figure 5. Facets generate counts for metadata categories and geographic regions in a GeoCMS

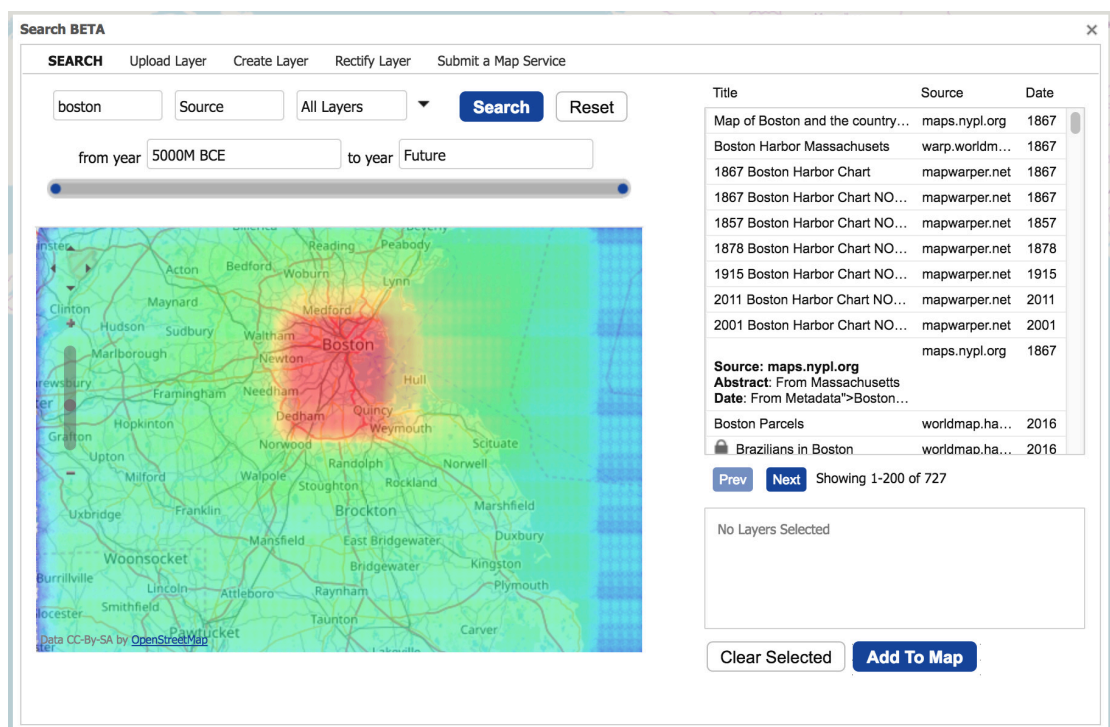


Figure 6. Spatial faceting enables heatmaps showing the distribution of the SDI layers in the space